

05/12/98 WED 08:50 FAX 503 264 3025

INTEL JF3-1

027

## INTEL INVENTION DISCLOSURE

INTEL CONFIDENTIAL

Legal ID **10155** (Legal use only)

DATE: 8/5/98

It is important to provide accurate and detailed information on this form (fill in ALL areas under Inventor(s)). The information will be used to evaluate your invention for possible filing as a patent application. When completed, please return this form to the Legal Department at HF3-03. If you have any questions regarding this form or to whom it should be forwarded, please call 756-1369 or 696-2851 or 554-3996. PLEASE, PRINT CLEARLY OR TYPE.

CEG (AZ) / CEG / CPD

## 1. Inventor(s):

Name Susan C. Kromenaker SS# [REDACTED]  
Empl. No. [REDACTED] Group/Div./Dept# [REDACTED] Ext [REDACTED] Mail Stop [REDACTED]  
Home Address [REDACTED]  
Citizenship USA Supervisor Scott Goble Ext [REDACTED] Mail Stop [REDACTED]  
Group Name CPD Software Division Name CPD

Name Mark L. Brown SS# [REDACTED]  
Empl. No. [REDACTED] Group/Div./Dept# [REDACTED] Ext [REDACTED] Mail Stop [REDACTED]  
Home Address see page #3  
Citizenship USA Supervisor Scott Goble Ext [REDACTED] Mail Stop [REDACTED]  
Group Name CPD Software Division Name CPD

Name Linda M. Roberts SS# [REDACTED]  
Empl. No. [REDACTED] Group/Div./Dept# [REDACTED] Ext [REDACTED] Mail Stop [REDACTED]  
Home Address see page #3  
Citizenship USA Supervisor Scott Goble Ext [REDACTED] Mail Stop [REDACTED]  
Group Name CPD Software Division Name CPD

Name William C. Arthur, Jr. SS# [REDACTED]  
Empl. No. [REDACTED] Group/Div./Dept# [REDACTED] Ext [REDACTED] Mail Stop [REDACTED]  
Home Address 14813 S. 24th St. Phoenix, AZ 85048  
Citizenship USA Supervisor Linda Roberts Ext [REDACTED] Mail Stop [REDACTED]  
Group Name CPD Software Division Name CPD

(PROVIDE SAME INFORMATION AS ABOVE FOR EACH ADDITIONAL INVENTOR)

2. Title of Invention: A Methodology to extend the I/O Architecture System Management Interface to allow a host processor to Gather and Modify performance monitor data collected by an I/O processor.

3. Stage of Development, i.e. % complete, and relation of technology to the following product/process: 95% complete, will be shipped on the Pro I/O Software Developer's kit in September, 1998.

4. (a) Has a description of your invention been, or will be shortly be, published?

NO            YES X DATE WAS OR WILL BE PUBLISHED September 1998

If YES, was the manuscript submitted for pre-publication approval? NO X YES           

(b) Has your invention been used/sold or planned to be used/sold by Intel or others?

NO            YES X DATE WAS OR WILL BE SOLD September 1998

5. If invention conceived, or constructed during performance of a government or third party contract, please check here            and give full contract Name and Number:           

6. Please attach a page to this form, DATED AND SIGNED BY ONE INVENTOR (PREPARER), to provide a brief summary of your invention and include the following information in your abstract.

(a) State general purpose(s) of your invention;

(b) Describe advantage(s) of your invention over what is done now;

(c) Describe essential elements of your invention; and

LEO299A/10-26-93 (BAY copy)

B.S.T.Z. DATABASE

INTEL CONFIDENTIAL

(d) Value of your invention of Intel (how will it be used?).

\*HAVE YOUR SUPERVISOR READ AND SIGN COMPLETED FORM.

DATE: 8/6/98 SUPERVISOR: Susan C. HammondDATE: 8/6/98 SUPERVISOR: Bob

BY THIS SIGNING, I (SUPERVISOR) ACKNOWLEDGE THAT I HAVE READ AND UNDERSTAND THIS DISCLOSURE, AND RECOMMEND THAT THE HONORARIUM BE PAID.

Enter Title in Summary Dialog B x

Page 2 of 4

INTEL CONFIDENTIAL

## INTEL INVENTION DISCLOSURE

# A Methodology to extend the I<sub>2</sub>O Architecture System Management Interface to allow a host processor to Gather and Modify performance monitor data collected by an I/O processor.

Susan C. Kromenaker, Mark L. Brown, Linda M. Roberts, William C. Arthur, Jr.  
8/6/98

## Purpose

This invention provides a method for a host processor to gather and modify data collected by an I/O processor containing a Performance Monitor unit. This invention extends the I<sub>2</sub>O Architecture by implementing private parameters on the IOP that can be extracted from the I/O Processor by the host, via the I<sub>2</sub>O Architecture System Management Interface.

## Advantages *host processor*

The I<sub>2</sub>O Architecture provides a method for gathering data from an I/O processor called the System Management Interface. Data is stored on the I/O Processor and extracted by the host processor by sending an *UtilParamsGet* message to the I/O Processor. The host is able to modify data stored on the I/O Processor by sending an *UtilParamsSet* message. Embedded within the messages is the exact data to modify and retrieve. This data is stored on the I/O Processor in parameter groups. For more information on the I<sub>2</sub>O Architecture, please review the *Intelligent I/O (I<sub>2</sub>O) Architecture Specification*, revision 1.5.

The newest Intel I/O processor, the 80960 RM and 80960 RN, contains a performance monitor unit. This unit gathers a variety of data that can be used for performance analysis, or for real-time system tuning. For more information, please review the *Tuzigoo/Huachuca Microprocessor EAS (Internal)*, revision 2.0. A separate patent disclosure has been filed for the performance monitor unit.

The I<sub>2</sub>O Architecture does not provide a method for *gathering* or *controlling* the I/O Processor's performance monitor hardware, nor for *gathering* data from the performance monitor hardware. This invention extends the I<sub>2</sub>O Architecture System Management interface and provides this capability.

By extending an industry standard such as the I<sub>2</sub>O Architecture to provide a method for gathering and modifying performance monitor data, Intel is providing a standard method of *configuring and controlling performance monitor hardware on an I/O Processor*. Intel is also providing a standard method of gathering performance monitor data from the I/O Processor. A software vendor developing tools to tune a server need only implement this method of configuring and controlling the Performance monitor hardware, and only implement this method of gathering performance monitor data from any I/O Processor that implements a performance monitor unit. No longer does the software vendor need to *tweak their tools* for each I/O processor.

A hardware vendor that implements a product utilizing an I/O Processor with a Performance Monitor Unit need only implement this method of letting the host control and configure the performance monitor hardware and gather performance monitor data. They are guaranteed to have their product work under any Operating System that implements the I<sub>2</sub>O Architecture, including NT, NetWare, SCO UNIX, HP-UX, OS/2 and Solaris. No OS changes are required. This significantly cuts down on development and maintenance costs.

## Key Elements

A Private Class called Performance Monitor has been defined. The methodology for defining a Private I/O Class is defined in the *Intelligent I/O (I<sub>2</sub>O) Architecture Specification*.

New parameter groups have been defined that contain the performance monitor data and performance monitor setup information. Software running on the host can modify parameters stored in these groups via the *UtilParamsSet* message. The type of information contained in these parameters is the information necessary for the I/O Processor to control and configure the Performance Monitor Hardware.

Software running on the host can gather performance monitor data in these new groups via the *UtilParamsGet* message. The type of data that is typically read from the parameters groups is the control and configuration information status, as well as the actual performance monitor results.

The exact parameter groups and their contents are defined in the *Performance Monitor Software API*, revision 1.1, which is attached to this document.

Enter Title in Summary Dialog Box

Page 3 of 4

**INTEL CONFIDENTIAL**

**Int I U**

CPD has already documented and implemented this invention. We plan to ship the associated software running on the I/O Processor in a Software Developer's Kit in September, 1998. We also plan to work with tool vendors to implement this methodology in their system performance monitoring tools.